(12) UK Patent Application (19) GB (11) 2 247 596 (13) A

(43) Date of A publication 04.03.1992

(21) Application No 9018794.9

(22) Date of filing 28.08.1990

(71) Applicant
Avesco Plc

(Incorporated in the United Kingdom)

Venture House, Davis Road, Chessington, Surrey, KT9 1TT, United Kingdom

(72) Inventor
Richard Conrad Jozefowski

(74) Agent and/or Address for Service
J P L Hooper
5 Haslingfield Road, Harlton, Cambridge, CB3 7ER, United Kingdom

(51) INT CL⁵
G09G 1/14

(52) UK CL (Edition K)
H4T TBEA

(56) Documents cited
GB 2236463 A GB 2223916 A EP 0240608 A2
EP 0146227 A2

(58) Field of search
UK CL (Edition K) H4T TBEA TBEX
INT CL⁵ G06F, G09G

(54) Anti-aliasing system

(57) An anti-aliasing system for use in the construction and manipulation of computer-generated video images comprises a frame store for holding the image data (sub-pixel data) at a resolution (Fig 2A) higher than that (Fig 2B) required for the display system; image data output means, taking the data from the frame store, converting it into a form having the lower resolution required by the display system, and in so doing using the stored sub-pixel data to determine the data in the relevant boundary display pixels, and outputting this lower resolution data to the display system.

The image data may be stored in a 1:1 bit mapped format, but is preferably compacted eg run-length encoded.

The displayed pixel values may depend on the weighted sums of the corresponding sub pixels (Figs 5A, 5B).

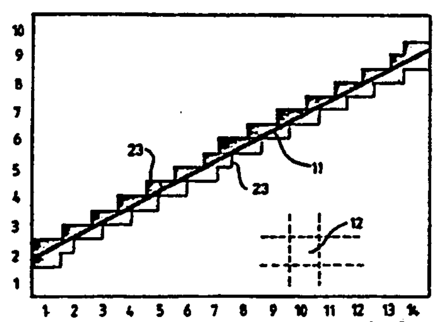

Fig.2A

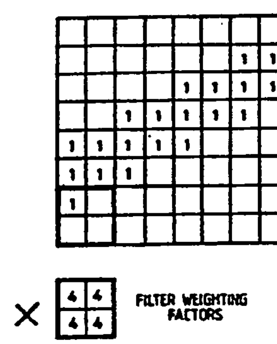

Fig. 5A

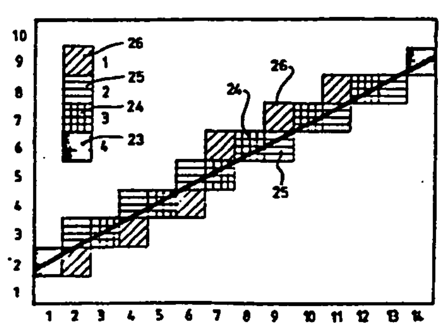

Fig. 2B

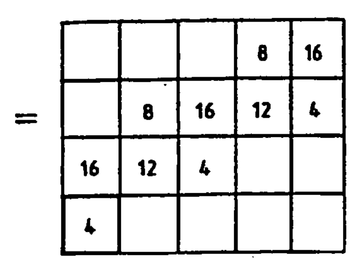

Fig.5B

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

The claims were filed later than the filing date within the period prescribed by Rule 25(1) of the Patents Rules 1990.

GB 2 247 596 A

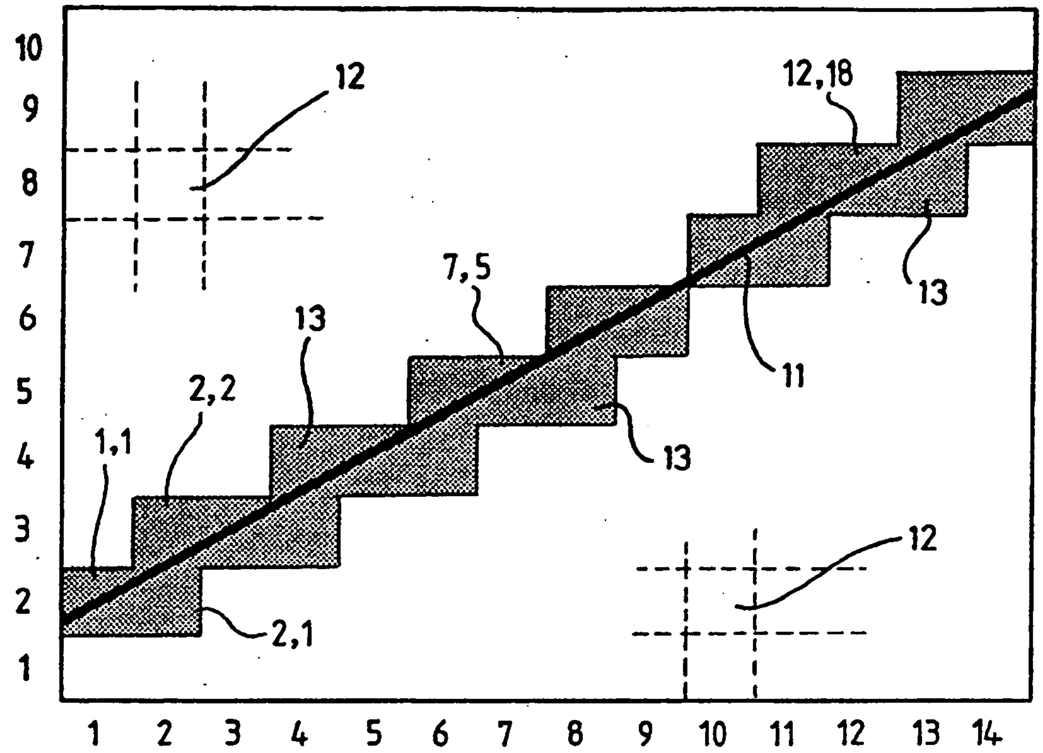
10
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14
12
12,18
7,5
13
11
2,2
1,1
2,1

Fig. 1A

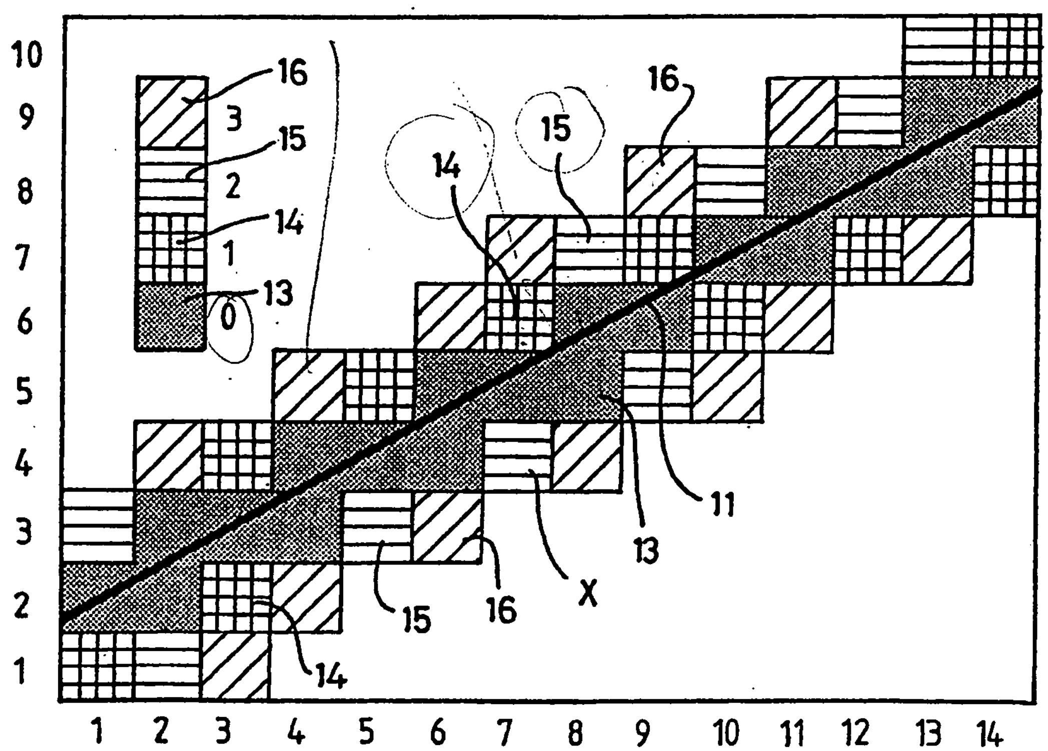
10
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14
16
3
15
2
14
1
13
0
11
X

Fig. 1B

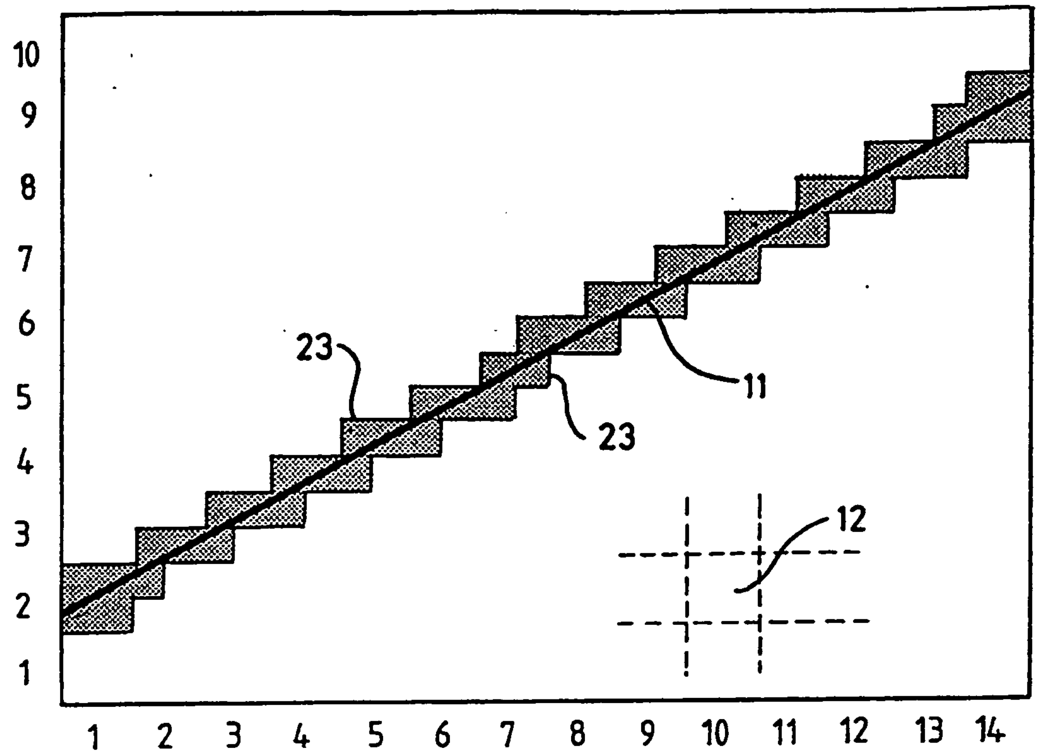
10
9
8
7
6
5
4
3
2
1
23
23
11
12
1 2 3 4 5 6 7 8 9 10 11 12 13 14

Fig. 2A

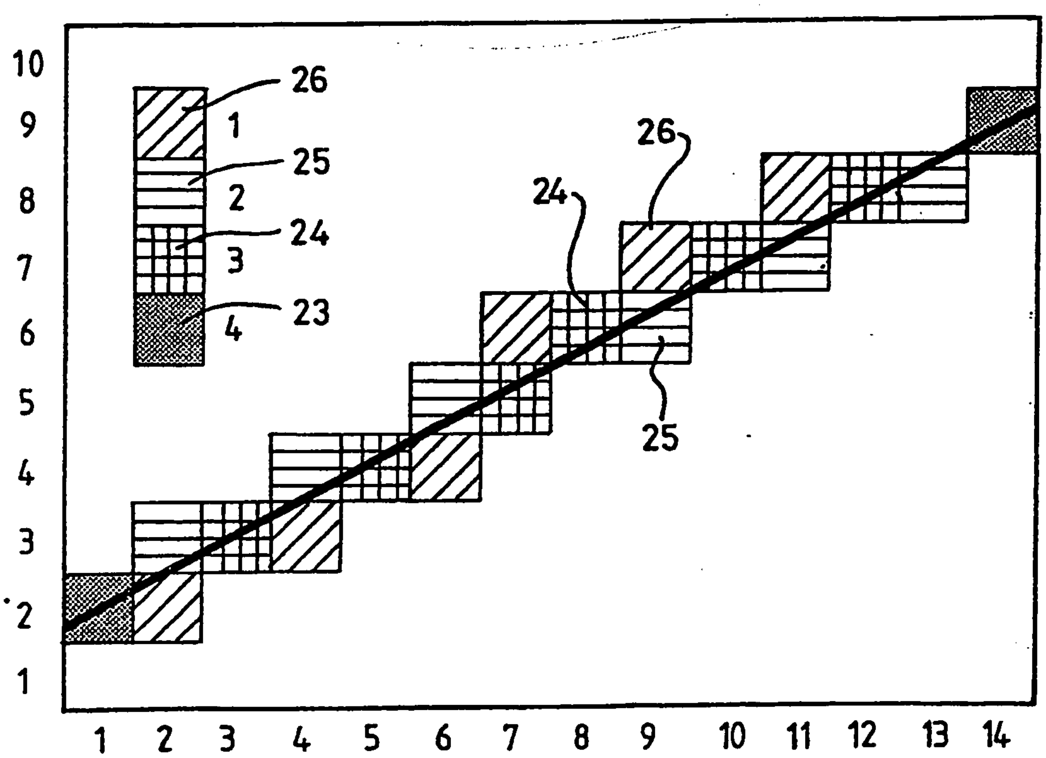
10
9
8
7
6
5
4
3
2
1
26
1
25
2
24
3
23
4
26
24
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14

Fig. 2B

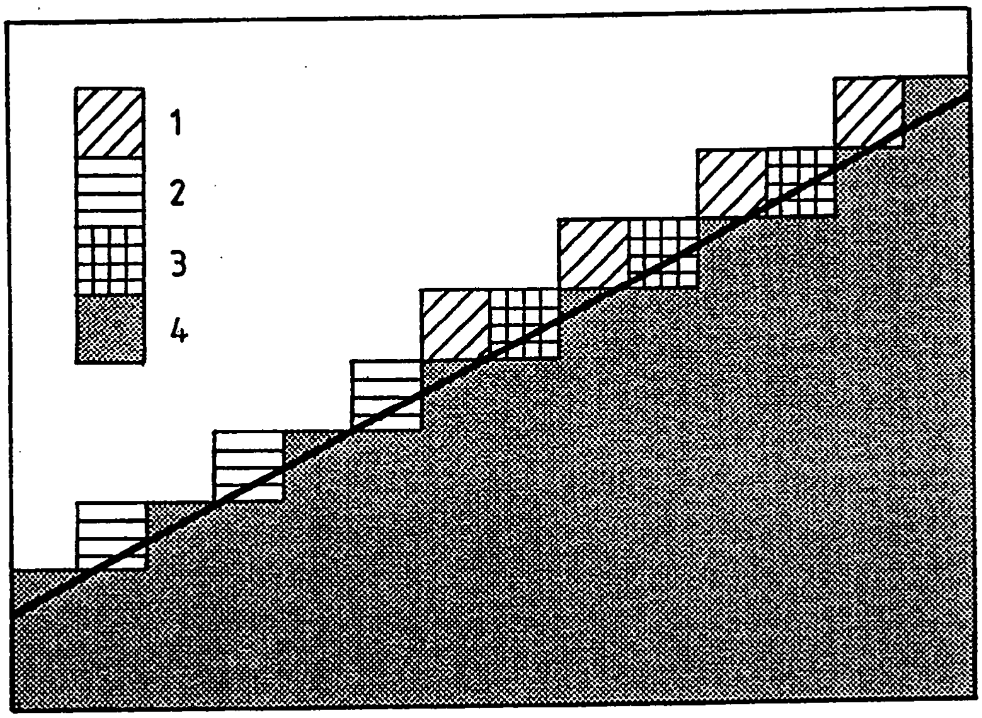

Fig. 2C

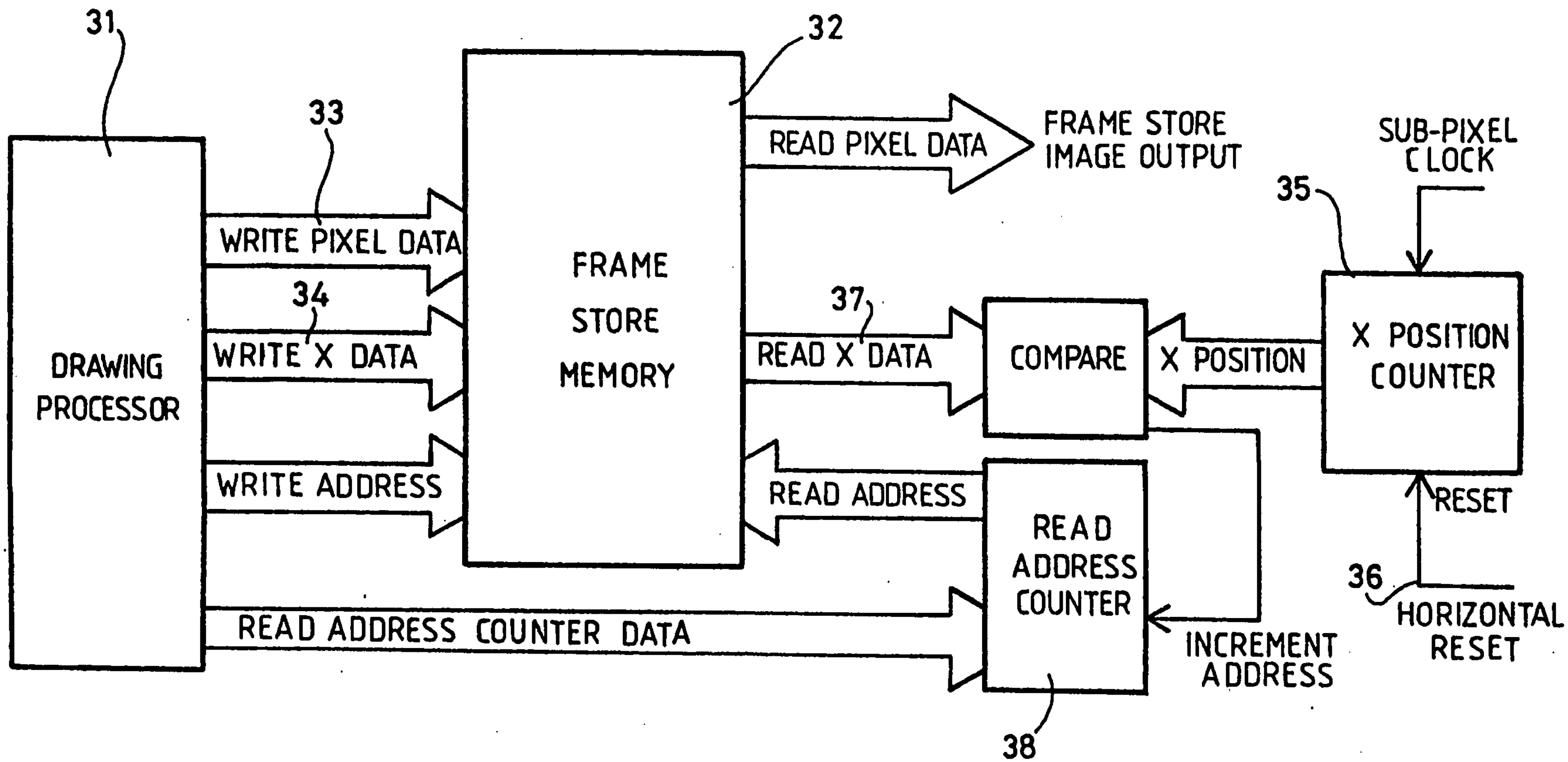
31
DRAWING
PROCESSOR
33
WRITE PIXEL DATA
34
WRITE X DATA
WRITE ADDRESS
READ ADDRESS COUNTER DATA
FRAME
STORE
MEMORY
32
READ PIXEL DATA
FRAME STORE
IMAGE OUTPUT
37
READ X DATA
COMPARE
READ ADDRESS
READ
ADDRESS
COUNTER
38
X POSITION
35
X POSITION
COUNTER
SUB-PIXEL
CLOCK
RESET
36
HORIZONTAL
RESET
INCREMENT
ADDRESS

Fig. 3

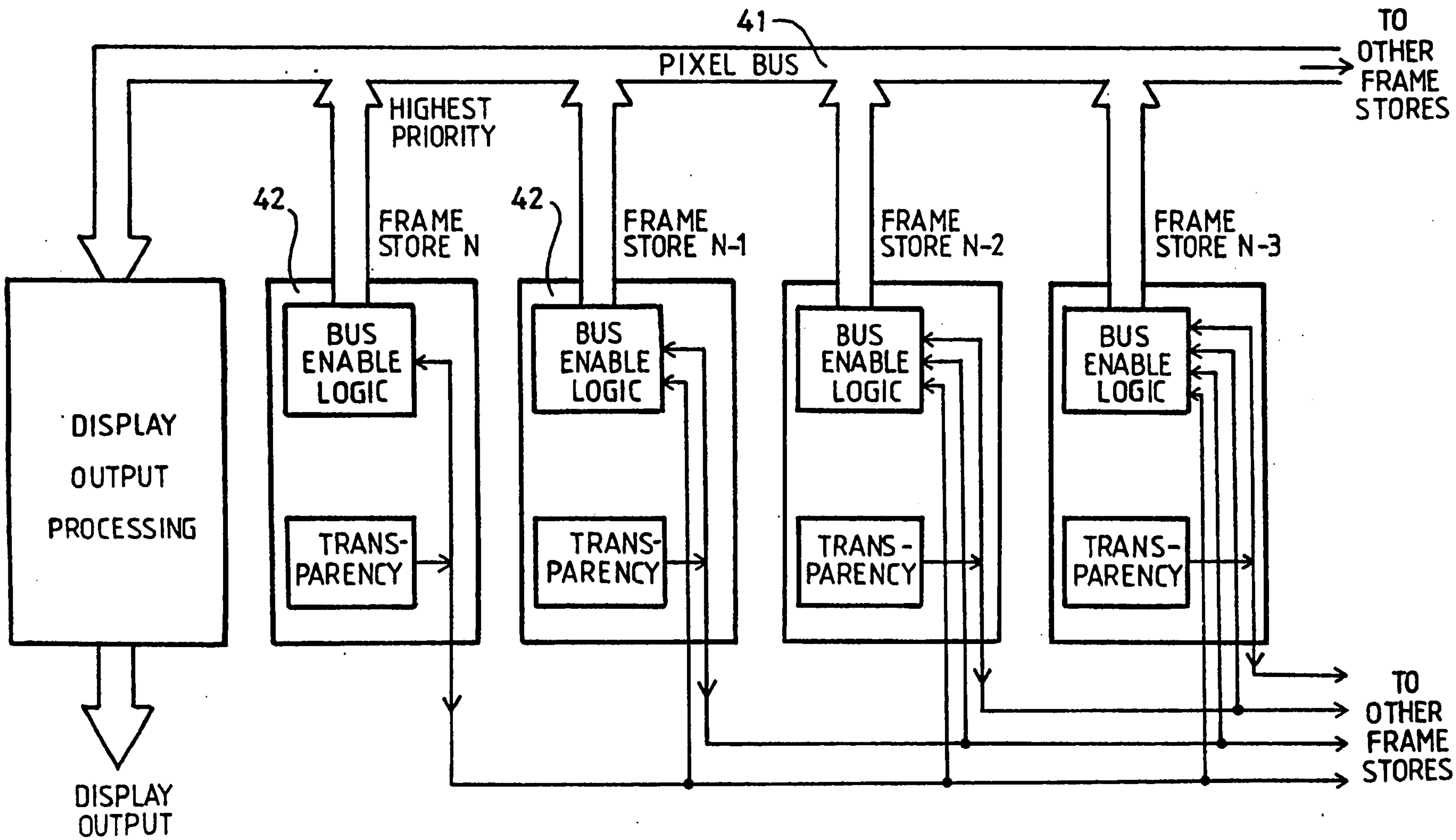
41
PIXEL BUS
TO OTHER FRAME STORES
HIGHEST PRIORITY
42
FRAME STORE N
42
FRAME STORE N-1
FRAME STORE N-2
FRAME STORE N-3
BUS ENABLE LOGIC
TRANS-PARENCY
BUS ENABLE LOGIC
TRANS-PARENCY
BUS ENABLE LOGIC
TRANS-PARENCY
BUS ENABLE LOGIC
TRANS-PARENCY
DISPLAY OUTPUT PROCESSING
DISPLAY OUTPUT
TO OTHER FRAME STORES

Fig. 4

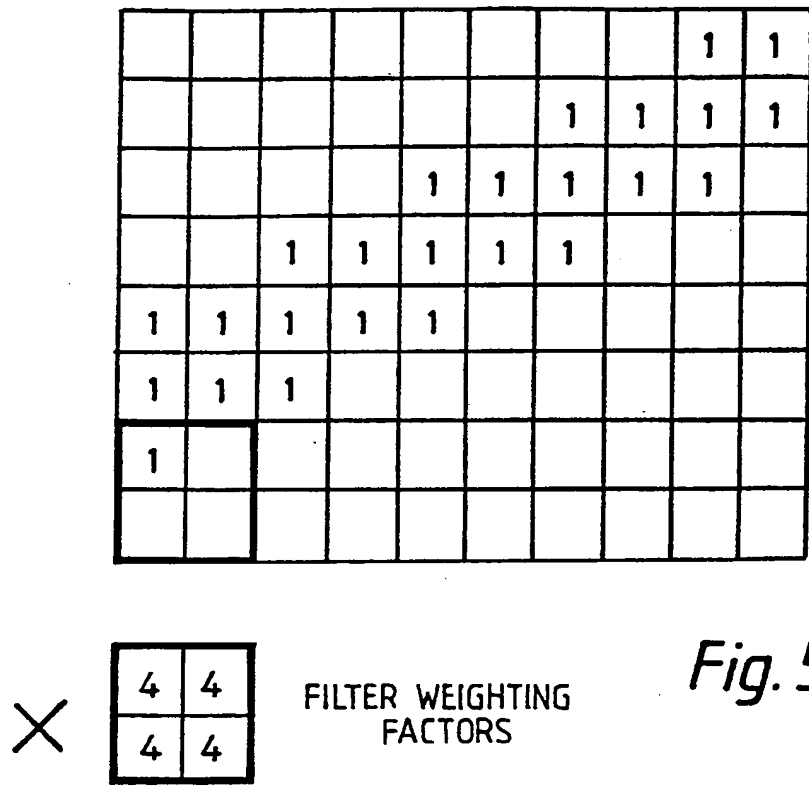

Fig. 5A

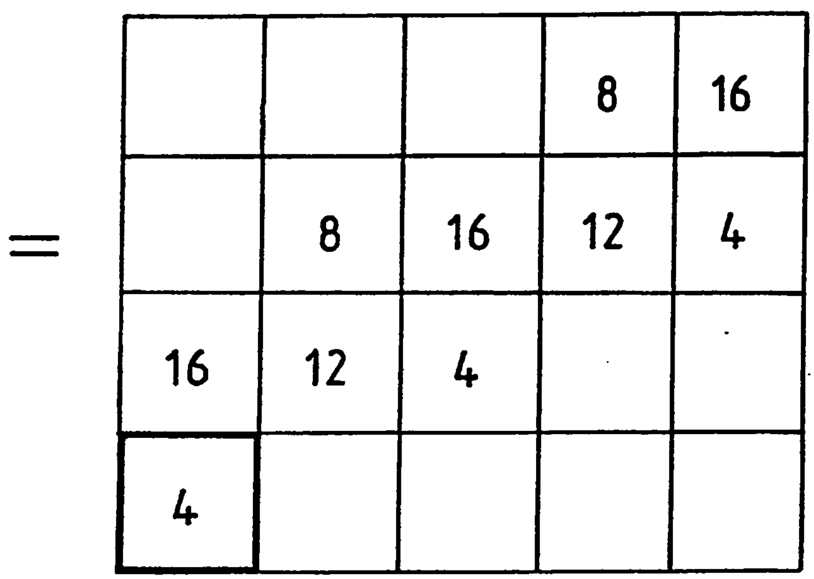

Fig. 5B

								1	1
						1	1	1	1
				1	1	1	1	1	
		1	1	1	1	1			
1	1	1	1	1					
1	1	1							
1									

×

1	2	1
2	4	2
1	2	1

FILTER WEIGHTING FACTORS

Fig. 6A

=

		1	6	13
1	6	13	13	6
13	13	6	1	
6	1			

Fig. 6B

Video Image Formation

This invention relates to video image formation, and concerns in particular the anti-aliasing of computer-generated video images.

It is now commonplace entirely to create and manipulate video - that is, television - images with a computer. At present these images tend to be rather simple in construction and filled with solid but rather flat expanses of colour,. but as processing techniques and capabilities improve so the images will become more natural and life-like, and be more complex, with depth and texture.

Apart from their visual quality, there are several problems associated with computer-generated video images. Firstly, while the basic idea of using digital frame store techniques to hold the image is now well established, whereby a single picture or object can be manipulated for the generation of special effects, nevertheless, in order for a computer system to draw multiple objects in real-time, such as might be required in a flight simulation system, vast processing power is required coupled with a very fast access frame store. Real-time in this case refers to a screen update rate which is, to the human viewer, sufficiently fast so as to be seen as continuous movement.

Secondly, there is the relatively severe difficulty of "aliasing" - the term used to identify the jagged step-ladder effect of a line (such as the edge of an object in the image) drawn at an angle across the screen obtained when the resolution of the generated image data

is higher than the computer system is capable of storing or the display system is capable of showing.

Processing power is improving, as computer technology provides ever faster and more complex processor units, and there are already a number of ways of encoding image data so as not only to enable the use of smaller frame stores but also to speed up their access. Improvements in display technology have provided visual display units with higher intrinsic resolutions, but as yet it is still difficult if not impossible to attain the sort of very high resolutions - 300 dots per inch and above - that are needed to convince the eye/brain that a line or edge is completely smooth and non-jagged. However, great strides have been made in subtlely modifying the displayed image so that a line or edge looks as though it is smooth and non-jagged even when it isn't - the concept is referred to as "anti-aliasing" - and the invention proposes another, and in some instances (at least) a superior, way of dealing with this difficulty.

The aliasing of a computer-generated image is a major problem, familiar to those involved in professional computer graphics. In any video display system the image is broken down into a number of rectangular picture elements, or pixels, horizontally and vertically. A typical PAL colour television frame, as used in most of Europe, is 625 lines deep; the top 50 lines are used for Teletext data, so that the actual picture image area is divided into about 720 such pixels horizontally and only 575 vertically; on a medium-sized screen this corresponds to a resolution of little better than 50 dots per inch. Each pixel is assigned a colour and intensity value which is uniform across its area.

It follows that a line which, drawn on paper, should be smooth and continuous, will, when drawn on the screen by lighting a sequence of pixels, be shown as jagged (stepped) and discontinuous if drawn at any angle other than horizontal or vertical (in these two special cases it can be represented by exactly aligned pixels close enough together to appear contiguous). The effect is know as "aliasing", and together with some techniques used to deal with it is described further hereinafter, especially with reference to the accompanying Drawings.

It will be appreciated, then, that with simple drawing techniques (as used in cartoon-type rather than real pictures), an image can appear as a mosaic of individual pixels, and geometric shapes, such as lines, will suffer from aliasing, appearing to have stepped edges. Technically, this aliasing is produced as a result of attempting to display an image at a higher resolution (with wider bandwidth) than the pixelisation process is capable of storing.

A number of techniques exist which aim to generate images in an anti-aliased form, thereby significantly improving the quality of the final display. These usually rely on drawing objects such that the boundary pixels (those around the edge of the image but not actually part of the image - particularly those adjacent the "risers" of any steps in the image edge) can take translucent values, so as to blend in with the image proper and make it look, to the eye, to be more realistic. This effect, too, is discussed hereinafter with reference to the accompanying Drawings.

In systems which use the translucency technique for anti-aliasing the necessary degree of translucency of each relevant boundary pixel is usually determined by estimating the proportion of that pixel which is

notionally covered by the ideal shape. To achieve this effect a read-modify-write process is required when accessing the frame store in which are held the values defining each pixel (this can be a software process, although for high speed hardware techniques are currently employed). First the system reads the old image information stored at the relevant location in the frame store, then it produces a modified version of that information taking into account the new image data that is going to overlay the old in the required (translucent) fashion, and then it writes back to the frame store this modified version in place of the original.

As might be expected, this technique has its own problems. Not only does it require a significant amount of hardware (in the form of multipliers and accumulators), but also it suffers from the fact that the final anti-aliased pixel lacks any sub-pixel information in the frame store to indicate which part of the pixel the object actually covers - that is to say, the amended version of the image data has completely replaced the original data, which is as a result lost, and any further manipulation of the image can only operate on the assumption that the amended version <u>is</u> the original version. In practice this will present problems when overlapping one object on another. For instance, a white circle on a black background will exhibit anti-aliasing components (the translucent boundary pixels) at various levels of grey. If this is now overlapped by a red circle of the same size anti-aliasing components should be formed which are various levels of red. However, because there is no sub-pixel information stored when the white circle is created - for the original black image data has been completely replaced by the <u>calculated</u> "white" image data - the red

image will contain anti-aliasing components at various levels of pink as the system blends the new, red, data with (what it now thinks is) the old, white, data.

Taking advantage of improvements in memory technology, whereby it is now feasible to construct frame stores with higher resolutions, the invention proposes a quite different technique for anti-aliasing, which technique is referred to hereinafter as "super sampling". Basically, the super sampling system maintains the stored image at a resolution that is significantly higher than that of the display system itself, and, at the output stage, converts this to the final display resolution. Using this method it is no longer necessary to calculate the degree of translucency of the relevant boundary pixels, or to use complex read-modify-write cycles to access the frame store, for the relevant boundary pixel effect can be determined upon the basis of the real sub0pixels that actually form part of the generated image, and the values in the relevant locations may be found by a simple read. Furthermore, this super sampling system maintains the correct sub-pixel information unchanged, which allows the outputs of one or more frame stores to be combined without degrading the final image.

In one aspect, therefore, the invention provides an anti-aliasing system for use in the construction and manipulation of computer-generated video images, in which system there is:

a frame store for holding the image data at a resolution higher than that required for the display system;

image data input means, for inputting the data to the frame store at the required high resolution; and

image data output means, for taking the data from the frame store, converting it into a form having the lower resolution required by the display system, and in so doing using the stored sub-pixel data to determine the data in the relevant boundary display pixels, and outputting this lower resolution data to the display system.

The anti-aliasing system of the invention employs a frame store which, in operation, will hold image data at a resolution that is higher than that at which the display system is capable of displaying the generated image. This frame store is itself perfectly conventional, and so may be any storage system used or suggested for use for this purpose. Normally, then, it will be constituted by one or more block of RAM - Random Access Memory - in semiconductor form together with its associated address means and so on. If, however, speed of operation is less important, then the frame store could be physically embodied in a magnetic (hard or floppy) disk or a read/write optical disk, or even a magnetic bubble memory device.

The image data held in the frame store could be stored in a higher-resolution version of the 1:1 bit mapping technique presently employed by the majority of frame stores used for professional graphics applications (in this basic bit-mapped technique each pixel on the screen has a corresponding memory location which describes its colour and intensity). However, because of the higher resolution used in the system of the invention means there is a very large amount of data to be held it is desirable to employ a more efficient method of storing the image. A number of algorithms exist for the compaction of data, especially image data,

which could be used here, but that felt most appropriate in the present case involves noting the changes in the picture, and then for each change storing merely the new value together with a number indicating how many pixels (or, in this case, sub-pixels) that value applies to. run length coding This not only results in a significant reduction in the amount of memory required to implement a frame store, but it also eliminates the need to fill solid objects. Such a compaction technique may conveniently be implemented as follows, assuming that the frame store is used to drive a raster display scanning horizontally downward (the process could be adapted for other mediums).

When creating an image, a drawing processor is used to sort the various objects making up the image. The sort is carried out in scan line order, taking account of overlapping objects, and an entry into the frame store memory is only made when a change in the image takes place. The entry consists of the colour and intensity value together with the horizontal position at which the change takes place. The display/drawing processor maintains the address into which data is placed in the frame store memory, and this address is incremented after writing each entry. To simplify display decoding, special entries may be coded to indicate control information, such as the end of a scan line.

run length decoding

When decoding the data for display. an X-position counter is used to indicate the current horizontal pixel position (this counter is reset by the horizontal reset signal at the start of each line, and is incremented for each sub-pixel position). When this counter value is equal to the read-X data output from the frame store memory, the display output is set to the corresponding

colour and intensity values, and the frame store read address counter is incremented so as to access the next entry. The read address counter determines what part of the frame buffer/store memory is used for display, and is usually set to the desired value by the drawing processor during the vertical blanking interval.

Whilst this principle is most efficient for dealing with the storage of solid objects having uniform colour and intensity, the frame store can be simply extended to store information regarding objects with uniform shaded surfaces. Effectively, this additional information is used to describe a colour gradient. At each horizontal pixel position, if there is no new entry in the frame store then the output remains the same as the previous value with the quite automatic addition of the colour gradient value.

Whether held in real or compacted form, the frame store holds the data defining every pixel - and, of course, sub-pixel - of the image. In a typical television application the frame store may allocate 32 bits of information for each of the sub-pixels. These are divided into eight bits for the red, green and blue components, with an additional eight bits to define key transparency. Naturally, the higher the resolution the larger the frame store will need in principle to be to hold all the information (although with compacted data the necessary size increase does not always match the resolution increase). However, from a practical point of view two factors are involved that either limit or enable the limitation of frame store size. Firstly, if the video image is to be maintained and up-dated as a real-time operation, which is highly desirable, then there is a fixed maximum time (of about 1/25 second for the relevant processor to draw the image into the frame

store - which means that there is little point in the frame store being larger (as regards the number of data entries it holds, and thus also as regards the image resolution) than can be drawn in during the available time. For a typical processor such as a Texas Instruments 320C30 operating at 13.5MHz this maximum store size is equivalent to about 16,000 entries.

Secondly, frame store size can also be limited because, using the system of the invention, it is possible to "divide" the generated image up into portions the data for which is stored independently and brought out and combined by a suitable number of processors acting in parallel. These portions, identified in some earlier stage of the image generation and processing, may each extend to cover the whole of the image area but at the same time represent what exists in a selected plane into the depth of the picture. Thus, there may be a background plane, which consists of little more than a coloured backdrop (and no matter how high the resolution only a small amount of data will be needed to define this). Similarly, the foreground plane may consist of a single object being manipulated in front of the background (and here, too, the amount of data will be small regardless of the resolution). And similarly yet again, there may be a number of intermediate planes behind the foreground plane but in front of the background plane, one in front of the next in an orderly sequence (the data for each of these needing only a small store). If, then, each plane requires no more than the maximum sized store, and each plane's store has its own processor working in parallel with all the other processors (perhaps under the control of some master processor), the whole image can be assembled within the available time span from more data

than could sequentially be manipulated satisfactorily in real time.

Accordingly, the image data can in principle be of any complexity, and held in any high resolution state, provided that it can be parcelled out to one or more processor/frame store combination each of which can process its data within the specified maximum time and all of which can be paralleled together to produce the complete image in the available time. This is discussed further hereinafter with reference to the accompanying Drawings.

Having said all that, in practice it is seldom necessary to hold the image data at more than four times the display resolution (four times requires 4 x 4 = 16 image sub-pixels per display pixel), and even twice the display resolution (requiring 2 x 2 = 4 sub-pixels per display pixel) is quite satisfactory, significant improvements being achieved over previous methods whereby the image is only maintained at the final display resolution (1 × 1 = 1).

One particular advantage of holding the image data at a higher resolution than the display data is that the original sub-pixel information is maintained no matter what happens to the relevant display pixel - and thus that the problem described above associated with an anti-aliased white-on-black circle overlaid with a red circle does not occur, for the new red data is not "measured" against the white-on-black data, to give pink, but against the still-present original black data. Thus, because the system maintains the sub-pixel information it is possible to overlap one object by another in an opaque manner, without having to worry about translucent pixel values to ensure correct anti-aliasing. Now, not only does this simplify the drawing

process, it also allows there to be overlapped objects generated by completely independent frame stores.

One application of this is that mentioned hereinbefore, the use of several frame stores to hold the image data relating to different depth planes of the image. Using a sequence of frame stores, each can be allocated a portion of the overall image to display (typically this partition is determined by another processor, which allocates objects, on the basis of distance from the viewing screen, to the various frame stores). The furthest objects from the screen are allocated to the first frame store. If a subsequent frame store contains an object at the same sub-pixel position then it replaces the value generated by the first store. This principle can be extended to any number of frame stores.

A typical implementation uses a common pixel data bus with prioritised frame stores so that only objects closest to the screen are permitted access to the bus. This is achieved by each frame store section "listening" to transparency information sent out by all the higher priority frame store sections. Only if all these higher priority sections are indicating transparency is the frame store permitted to place data on to the pixel bus.

Such a system utilises one pixel bus, and so does not allow for objects to be translucently overlaid on to objects generated by low priority frame stores. However, this translucent overlay can be achieved by an extension to the system, using a "daisy chain" arrangement where the output of the first frame store is fed into the second frame store and the objects contained in the second frame store can obscure the objects contained in the first by varied levels of translucency, controlled by transparency information in

the second frame store. The resultant output from the second frame store is fed into the third frame store and so on up to any number of frame stores.

The system of the invention includes means to input data to the (or each) frame store, as well as means to output that data from the frame store, _via_ a resolution converter, to the display system. The input means may be any of the type used for placing (computer-generated) image data into a frame store, with the proviso that it should include, or be able to handle the data from, the desired data compaction equipment. This type of input means is generally well known, and needs no further comment here, but even so the following comments may be helpful.

The frame store will normally be a block of RAM, and ideally this will be dual-ported, with completely separate and independent means of addressing individual memory locations for the input and output functions. Input of data will typically be _via_ the data bus of a micro-processor, with the frame store memory being directly mapped into its address space. With dual-ported memory it is therefore possible to connect the input port directly to the processor's address and data busses. Each entry in the frame store signifies a change in the image to be displayed. These changes are stored in memory in the order in which they occur, and it is the responsibility of the processor system to load the frame store in this order. An entry comprises a colour/intensity value together with an offset idicating at what position on the screen this data will take effect. Special entries may also be used to encode control information to assist decoding.

The output means must not only take image data from the frame store (and so include, or be able to send the

data to, data decompaction means that can resurrect the original image data from its stored compacted form), it must also be able to convert the data from its stored high resolution form into the lower resolution form necessary for the display system. The high speed required will normally necessitate the use of purpose-designed hardware. This hardware will maintain an address counter to point to the next output location in the frame store. When a change occurs in the image, the data output from the frame store is used to update the image, and the hardware address counter is incremented, so as to point to the next location in the frame store. Provision will usually be made to preset the address counter to a known value at the start of each frame, this typically being under the control of the processor system.

As explained, each display pixel corresponds to a number of sub-pixel elements in the image data (a simple system which maintains the frame store at twice the horizontal and vertical display resolution would have four sub-pixels for every display pixel), and these must be combined in a suitable manner to obtain an acceptable image. Intuitively it might be expected that simple averaging of these four pixels would provide a reasonable result, and indeed this is so (described hereinafter with reference to the accompanying Drawings is the case of a straight line, super-sampled at twice the horizontal and vertical display resolutions, fed through a simple filter to give the resultant display values). In practice more complicated weightings of the four sub-pixels, together with a number of adjacent sub-pixels, will yield significantly better results. Effectively this can be regarded as acting as a two dimensional FIR (finite impulse response) filter, with the weighting factors used to tailor the desired impulse

response, and again such a case is described hereinafter with reference to the accompanying Drawings. Even more sophisticated techniques can be used which involve adding a proportion of previous output pixel values surrounding the new pixel value. This can be regarded as a two dimensional recursive or IIR (infinite impulse response) filter. The precise filter characteristic used will normally depend upon technical and cost restraints.

The anti-aliasing system of the invention is intended to form part of a complete image generating and manipulating arrangement, and as such it will be employed as the heart of a collection of hardware having at its input end operator-controlled data-input devices such as touch tablets or screen, mice, joysticks, video cameras, text font data stores, and the like, and at its output end display devices such as video screens, printers, films and so forth. These aspects are well known in the Art, and need no further comment here.

It will be appreciated that the invention employs a number of techniques which enable solid objects to be drawn very rapidly by a relatively inexpensive processor. Furthermore, it enables a very effective method for partitioning the processor load in a multi-processor system. The novel aspects of the invention may be summarised as follows:-

1) The storage of the image data at a higher resolution than that ultimately required for the display medium.

2) The holding of the data in the frame store in a compacted form.

3) The ability to combine the output of any number of frame stores without any degradation in the quality of the final image.

4) The conversion of the high resolution stored image to the lower resolution required by the display medium.

It should also be observed that whilst the system has been described as though the image were to be directly displayed on a television-type screen, it is equally applicable to other output/display mediums, such as film, and to intermediate storage devices, such as memory and magnetic or optical disks.

Various embodiments of the invention are now described, though by way of illustration only, with reference to the accompanying Drawings in which:

<u>Figure 1A</u> shows an extremely exaggerated graphical example of aliasing of a line displayed at a lower resolution than it is generated;

<u>Figure 1B</u> shows the application to the line of Figure 1A of one type of anti-aliasing technique;

<u>Figure 2A</u> shows a graphical example of a line like that of Figure 1A actually stored at twice the display resolution - super sampled in accordance with the invention - and then as it would be were it to be displayed at that resolution, without any anti-aliasing;

<u>Figure 2B</u> shows the line of Figure 1A now displayed at the normal display resolution and with anti-aliasing in accordance with the invention;

<u>Figure 2C</u> shows an edge like the line of Figure 1A, displayed at the normal display resolution and with anti-aliasing in accordance with the invention;

<u>Figure 3</u> shows a simplified block diagram of the components for utilising a compacted image data frame store;

Figure 4 shows a simplified block diagram of the components for utilising several image data frame stores in parallel;

Figure 5A shows a numeric representation of a section of a super sampled line like that of Figure 2A;

Figure 5B shows the same numeric representation after weighted "filtering" to convert the stored values to display values; and

Figures 6A & B show numeric representations similar to those of Figures 5A & B, but using different "filtering" weighting factors.

Figure 1A is an extremely exaggerated illustration of the aliasing effect. It shows a line (11) drawn, at very high resolution, across a screen that is capable of showing nothing better than one centimetre square block (as 12), or picture element (pixel) per centimetre (a resolution of about 2.5 "dots" per inch). The Figure comprises an array of such blocks, 14 across (the X-coordinate) and 10 up (the Y-coordinate). Wherever the real line 11 crosses or touches a block that block is lit (shown dark in the Figure). Thus, where the real line touches the blocks (as 13) at coordinates (1,1), (2,1), (2,2), ... (7,5), ... (12,8) and so on, so these blocks are shown blacked in in the Figure.

It will be apparent that the line as actually displayed - the collection of blacked-in blocks 13 - is jagged and stepped, though the real line is smooth and step-free; the displayed line is jagged - aliased -

because the resolution of the display system (the blocks) is so low. Were the display resolution to increase - were the size of the blocks to decrease, and their number per inch to increase - the line would, to the eye, become steadily less and less jagged (see the similar line at twice the resolution in Figure 2A), until at around 250 blocks per inch (or roughly 100 blocks per centimetre) it would appear perfectly smooth when seen from the normal viewing distance of a couple of feet (60 cm).

Figure 1B shows the application of one type of anti-aliasing technique to the same line, at the same very low resolution, in an attempt to make it look smooth even when it is in fact stepped and jagged. This particular anti-aliasing technique lights (fills in) various blocks adjacent to the ones through which the line actually passes. The latter are shown fully filled (as 13), while blocks which are near the line but not actually crossed thereby are shown hatched in different ways, the degree of hatching matching the closeness to the line (those closest are cross-hatched, as 14, those a little further away are horizontally hatched, as 15, and those still further away are diagonally hatched, as 16). As shown, this hatching is purely diagrammatic, and is intended merely to represent the sort of effect one might obtain by making the boundary pixels - the blocks near the real line - take on a "mixture" of the line pixel characteristics and the underlying background pixel characteristics. In this case this is shown as degrees of "grey".

If Figures 1A and 1B are viewed from about 15 feet (5 metres) the aliased line of Figure 1A will clearly be

seen to be jagged, while the anti-aliased line of Figure 1B will seem to be smoother (albeit slightly "fuzzy"). In this way can the aliasing problem be at least partially overcome.

In a real situation the degree of translucency is usually determined by estimating the proportion of the pixel which is covered by the ideal shape. To achieve this effect a read-modify-write process is required when accessing the frame store. However, not only does this require a significant amount of hardware but also it suffers from the fact that the final anti-aliased pixel lacks any sub-pixel information in the frame store to indicate which part of the pixel the object actually covers. Looking at Figure 1B, and considering block X at coordinates (7,4), it will be apparent that a block which was originally unlit (white), because the line 12 did not actually go through it, has become, after the anti-aliasing process has been applied, lit to the medium grey level as shown, and that this new and "imitation" data has been stored in the frame store, overwriting the real data that was originally there. This will cause problems if that same block is again involved in the anti-aliasing process, because now the relevant frame store location no longer holds the correct data on which the anti-aliasing procedure should be working.

The invention uses an alternative technique for anti-aliasing called, for convenience, super sampling. Using a frame store with a higher than normal resolution, the technique maintains the image at this higher resolution and, at the output stage, converts this to the final, lower, display resolution. Maintaining the stored data at the higher, sub-pixel,

resolution means that the original data may be used to calculate and send to the display the pixel values for the display, including the boundary "translucent" values, without changing the real values held in the frame store.

Figure 2A illustrates a line 11 similar to that shown in Figure 1A but stored at - and here drawn purely for illustrative purposes at - twice the display resolution horizontally and vertically. Figure 2B shows at that lower resolution the display output one could expect using simple filtering techniques. Just as in Figure 1A, the Figure 2A line 11 "lights" the sub-pixels (as 23) it actually passes through, and even at a resolution twice that of Figure 1A (2 blocks per centimetre instead of 1) the Figure 2A line is still stepped and jagged. Figure 2B shows the same line as actually seen on a 1 block per centimetre display, with anti-aliasing according to the invention. Here the sub-pixel data in the frame store has been used to calculate the "transparency" of each display pixel, so that where a pixel contains four lit sub-pixels it is filled in black (as 23), where it contains three it is filled in cross-hatched (as 24), and where it contains two or one it is filled in horizontally hatched or diagonally hatched respectively (as 25 and 26).

Figure 2C shows an edge (rather than a line) as stored and displayed in the same way.

The actual resolution at which the image data is maintained is a compromise between the size of the memory array and the accuracy of the anti-aliasing. Whatever it is, though, this super sampling technique means it is not necessary to calculate the degree of translucency, nor need complex read-modify-write cycles be employed to access the frame store.

In the system of the invention it is desirable to store the image data in a compacted form. This not only results in a significant reduction in the amount of memory required to implement a frame store but it also eliminates the need to fill solid objects.

The following description refers to the simplified block diagram, Figure 3, and assumes that the frame store is used to drive a raster display scanning horizontally downward and having a horizontal blanking interval at the end of each horizontal scan and a vertical blanking interval at the end of each frame. The process could, however, be adapted for other mediums.

When creating an image, a drawing processor (31) is used to sort the various objects it is dealing with. The sort is carried out in scan line order, taking account of overlapping objects, and an entry into the frame store memory (32) is only made when a change in the image takes place. The entry consists of the colour and intensity value (sent via the write pixel data bus, 33), together with the horizontal (X) position at which the change takes place (sent via the write X data bus 34). The drawing processor 31 maintains the address into which data are placed in the frame store memory 32, and this address is incremented after writing each entry. To simplify display decoding, special entries may be coded to indicate control information, such as the end of a scan line.

When decoding the data for display. the X-position counter (35) is used to indicate the current horizontal pixel position. The counter is reset by the horizontal reset signal (on line 36) at the start of each line, and is incremented for each sub-pixel position. When this counter value is equal to the read X-data output (on

line 37) from the frame store memory 32, the display output is set to the corresponding colour and intensity values, and the frame store read address counter (38) is incremented so as to access the next entry. The read address counter 38 determines what part of the frame store memory 32 is used for display, and is usually set to the desired value by the drawing processor 31 during the vertical blanking interval.

The simplified block diagram of Figure 3 shows the frame store memory 32 as being dual ported - ie it has separate read and write buses. In practice, a common bus may be used which is time multiplexed between read and write functions.

It should be noted that, although it is not here shown specifically, because the frame store maintains the image at a higher vertical resolution than that of the output device, this part of the circuit is duplicated according to the ratio of the two resolutions. Thus, if the frame store is at twice the vertical display resolution, two similar circuits will be employed so as simultaneously to generate two output values along the length of the scan line (raster).

Figure 4 deals with the combination of several frame store outputs. The inventive system maintains correct sub-pixel information which allows one to combine the outputs of one or more frame stores without degrading the final image.

As the frame store maintains an image to a higher resolution than that of the final display device, each pixel will contain a number of sub-pixel elements. For instance, if the frame store is at twice the horizontal and vertical resolutions then each pixel will comprise

four sub-pixels. In a typical television application the frame store may allocate 32 bits of information for each of the sub-pixels. These are divided into eight bits for the red, green and blue components with an additional eight bits to define key transparency. Because the system maintains this sub-pixel information it is possible to overlap one object by another in an opaque manner, without having to be concerned with translucent pixel values to ensure correct anti-aliasing. Not only does this simplify the drawing process, it also allows the overlapping of objects generated by completely independent frame stores.

Each frame store can be associated with a portion of the overall image to display. Typically, this partition is determined by another processor which allocates objects, on the basis of distance from the viewing screen, to the various frame stores. The furthest objects from the screen are allocated to the first frame store. If a subsequent frame store contains an object at the same sub-pixel position then it replaces the value generated by the first store. This principle can be extended to any number of frame stores.

A typical implementation, shown in Figure 4, uses a common pixel data bus (41) with prioritised frame stores (as 42) so that only objects closest to the screen are permitted access to the bus. This is achieved by each frame store section 42 "listening" to transparency information sent out by all the higher priority frame store sections. Only if all these higher priority sections are indicating transparency is each frame store permitted to place data on to the pixel bus.

The simple system shown in Figure 4 utilises one pixel bus 41, and does not allow for objects to be translucently overlaid on to stores generated by low

priority frame stores. This can be achieved by an extension to this system using a "daisy chain" arrangement where the output of the first frame store is fed into the second frame store and the objects contained in the second frame store can obscure the objects contained in the first by varied levels of translucency, controlled by transparency information in the second frame store. The resultant output from the second frame store is fed into the third frame store and so on up to any number of frame stores.

The final aspect of the invention involves conversion of the frame store resolution to that required by the display device. This is described with reference to Figures 5A & B and Figures 6A & B.

As explained, each display pixel comprises a number of sub-pixel elements. These must therefore be combined in a suitable manner to obtain an acceptable image. A simple system which maintains the frame store at twice the horizontal and vertical display resolution would therefore have four sub-pixels for every display pixel. Simple averaging of these four pixels provides a reasonable result, as shown in Figure 5.
Figure 5A shows the stored numeric values representing part of a thick straight line, super-sampled at twice the horizontal and vertical display resolutions. Each store (sub-pixel) location is holding the value "1". These are fed through a simple filter, whereby the sub-pixel location values are each multiplied by 4, the four results are added together, and the resultant sum is used to define the display pixel value.

It will be seen that in this way the four sub-pixels (only one of which is of value 1 in the bottom

left-hand corner (coordinates 1, 1) of Figure 5A are converted to

$$1\times4 + 0\times4 + 0\times4 + 0\times4 = 4$$

which is the value for the corresponding pixel (at 1, 1) in Figure 5B, while the four sub-pixels each of value 1 immediately above that one (at 1, 2 in Figure 5A) are converted to

$$1\times4 + 1\times4 + 1\times4 + 1\times4 = 16$$

for the pixel at 1, 2 in Figure 5B. Similarly, the four sub-pixels three of which are of value 1 to the right of the latter (at 2, 2) are converted to

$$1\times4 + 1\times4 + 1\times4 + 0\times4 = 12$$

In practice more complicated weightings of the four sub-pixels, together with a number of adjacent sub-pixels, will yield significantly better results. Effectively this can be regarded as acting as a two dimensional FIR (finite impulse response) filter, with the weighting factors used to tailor the desired impulse response. Figures 6A & B show the effect of such a slightly more complicated filter on the display output values. The filter weighting factors are as shown - they are 2, 4, 1 and 2 for the four sub-pixels actually belonging to this particular pixel, and 1, 2, 1, 2 and 1 for the five sub-pixels above and to the right (as viewed). Placing the filter over the bottom left-hand pixel of Figure 6A (at 1, 1), the sum of all nine weighted sub-pixel values is

$$1\times1 + 1\times2 + 1\times1 + 1\times2 + 0\times4 + 0\times2 + 0\times1 + 0\times2 + 0\times1 = 6$$

which is the value inserted into the corresponding display pixel in Figure 6B. Similarly, the weighted values for the display pixels at 1, 2 and 1, 3 are

$$0\times1 + 0\times2 + 1\times1 + 1\times2 + 1\times4 + 1\times2 + 1\times1 + 1\times2 + 1\times1 = 13$$

and

$$0\times1 + 0\times2 + 0\times1 + 0\times2 + 0\times4 + 0\times2 + 0\times1 + 0\times2 + 1\times1 = 1$$

respectively.

Even more sophisticated techniques can be used which involve adding in a proportion of previous output pixel values surrounding the new pixel value. This can be regarded as a two dimensional recursive or IIR (infinite impulse response) filter. The precise filter characteristic used will normally depend upon technical and cost restraints.

CLAIMS

1. An anti-aliasing system for use in the construction and manipulation of computer-generated video images, in which system there is:

a frame store for holding the image data at a resolution higher than that required for the display system;

image data input means, for inputting the data to the frame store at the required high resolution; and

image data output means, for taking the data from the frame store, converting it into a form having the lower resolution required by the display system, and in so doing using the stored sub-pixel data to determine the data in the relevant boundary display pixels, and outputting this lower resolution data to the display system.

2. An anti-aliasing system as claimed in Claim 1, wherein the image data frame store is constituted by one or more block of RAM.

3. An anti-aliasing system as claimed in either of the preceding Claims, wherein in use the image data is held in the frame store in a compacted form.

4. An anti-aliasing system as claimed in Claim 3, wherein in use the frame store drives a raster display scanning horizontally downward, and there is provided a drawing processor used to sort the various objects making up the image, this being carried out in scan line order, taking account of overlapping objects, and an entry into the frame store memory is only made when a change in the image takes place, the entry consisting of at least the colour and intensity value together with

the horizontal position at which the change takes place, the drawing processor maintaining the address into which data is placed in the frame store memory, this address being incremented after writing each entry,

and wherein for decoding the data for display, an X-position counter is used to indicate the current horizontal pixel position, and when this counter value is equal to the read-X data output from the frame store memory, the display output is set to the corresponding colour and intensity values, and the frame store read address counter is incremented so as to access the next entry.

5. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store allocates 32 bits of information for each of the sub-pixels.

6. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store size is sufficient for 16,000 entries.

7. An anti-aliasing system as claimed in any of the preceding Claims, wherein there is a multiplicity of frame stores, and associated processors, acting in parallel.

8. An anti-aliasing system as claimed in Claim 7, wherein several frame stores are used to hold the image data relating to different depth planes of the image, and there is provided another processor which allocates objects, on the basis of distance from the viewing screen, to the various frame stores, the furthest objects from the screen being allocated to the first frame store.

9. An anti-aliasing system as claimed in Claim 8, which uses a common pixel data bus with prioritised frame stores so that only objects closest to the screen

are permitted access to the bus, each frame store section "listening" to transparency information sent out by all the higher priority frame store sections, and only being permitted to place data on to the pixel bus if all these higher priority sections are indicating transparency.

10. An anti-aliasing system as claimed in any of the preceding Claims, wherein the image data is held at four times the display resolution.

11. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store is a block of dual-ported RAM, with separate and independent means of addressing individual memory locations for the input and output functions, and input of data is <u>via</u> the data bus of a micro-processor, with the frame store memory being directly mapped into its address space, the input port being directly connected to the processor's address and data busses.

12. An anti-aliasing system as claimed in any of the preceding Claims, wherein the output means is a hardware device which maintains an address counter to point to the next output location in the frame store, and includes means to preset the address counter to a known value at the start of each frame, under the control of the processor system.

13. An anti-aliasing system as claimed in any of the preceding Claims, wherein each display pixel corresponds to a weighted average of a number of sub-pixel elements in the image data.

14. An anti-aliasing system as claimed in any of the preceding Claims and substantially as described hereinbefore.

15. An image generating and manipulating arrangement incorporating an anti-aliasing system as claimed in any of the preceding Claims.

16. A process for the anti-aliasing of computer-generated video images, in which:

the image data is stored, optionally in compacted form, at a higher resolution than that ultimately required for the display medium; and

prior to use the stored data is then converted to the lower resolution required by the display medium.

17. A process as claimed in Claim 16 and substantially as described hereinbefore.

Amendments to the claims
have been filed as follows

1. An anti-aliasing system for use in the construction and manipulation of computer-generated video images, in which system there is:

frame store means for holding the image data at a resolution higher than that required for the display system;

image data input means, for inputting the data to the frame store at the required high resolution; and

image data output means, for taking the data from the frame store, converting it into a form having the lower resolution required by the display system, and in so doing using the stored sub-pixel data to determine the data in the display pixels, and outputting this lower resolution data to the display system;

in which system:

there is a multiplicity of frame stores, each with its associated processor(s), acting in parallel, and these frame stores are used to hold the sub-pixel image data relating to different depth planes of the image; and

the image data output means effects the combination and conversion of all the output sub-pixel image data to determine the data in the display pixels without degrading the sub-pixel data in each frame store.

2. An anti-aliasing system as claimed in Claim 1, wherein the image data frame store is constituted by one or more block of RAM.

3. An anti-aliasing system as claimed in either of the preceding Claims, wherein in use the image data is held in the frame store in a compacted form.

4. An anti-aliasing system as claimed in Claim 3, wherein in use the frame store drives a raster display scanning horizontally downward, and there is provided a drawing processor used to sort the various objects making up the image, this being carried out in scan line order, taking account of overlapping objects, and an entry into the frame store memory is only made when a change in the image takes place, the entry consisting of at least the colour and intensity value together with the horizontal position at which the change takes place, the drawing processor maintaining the address into which data is placed in the frame store memory, this address being incremented after writing each entry,

and wherein for decoding the data for display, an X-position counter is used to indicate the current horizontal pixel position, and when this counter value is equal to the read-X data output from the frame store memory, the display output is set to the corresponding colour and intensity values, and the frame store read address counter is incremented so as to access the next entry.

5. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store allocates 32 bits of information for each of the sub-pixels.

6. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store size is sufficient for 16,000 entries.

7. An anti-aliasing system as claimed in any of the preceding Claims, wherein several frame stores being used to hold the image data relating to different depth planes of the image, there is provided another processor

which allocates objects, on the basis of distance from the viewing screen, to the various frame stores, the furthest objects from the screen being allocated to the first frame store.

8. An anti-aliasing system as claimed in Claim 7, which uses a common pixel data bus with prioritised frame stores so that only objects closest to the screen are permitted access to the bus, each frame store section "listening" to transparency information sent out by all the higher priority frame store sections, and only being permitted to place data on to the pixel bus if all these higher priority sections are indicating transparency.

9. An anti-aliasing system as claimed in any of the preceding Claims, wherein the image data is held at four times the display resolution.

10. An anti-aliasing system as claimed in any of the preceding Claims, wherein the frame store is a block of dual-ported RAM, with separate and independent means of addressing individual memory locations for the input and output functions, and input of data is <u>via</u> the data bus of a micro-processor, with the frame store memory being directly mapped into its address space, the input port being directly connected to the processor's address and data busses.

11. An anti-aliasing system as claimed in any of the preceding Claims, wherein the output means is a hardware device which maintains an address counter to point to the next output location in the frame store, and includes means to preset the address counter to a known value at the start of each frame, under the control of the processor system.

12. An anti-aliasing system as claimed in any of the preceding Claims, wherein each display pixel corresponds to a weighted average of a number of sub-pixel elements in the image data.

13. An anti-aliasing system as claimed in any of the preceding Claims and substantially as described hereinbefore.

14. An image generating and manipulating arrangement incorporating an anti-aliasing system as claimed in any of the preceding Claims.

15. A process for the anti-aliasing of computer-generated video images, in which:

the image data is stored, optionally in compacted form, at a higher resolution than that ultimately required for the display medium, the storage being in a multiplicity of frame stores, each with its associated processor(s), acting in parallel, these frame stores being used to hold the sub-pixel image data relating to different depth planes of the image; ; and

prior to use the stored data is then converted to the lower resolution required by the display medium, this being effected without degrading the sub-pixel data in each frame store.

16. A process as claimed in Claim 15 and substantially as described hereinbefore.

Published 1992 at The Patent Office. Concept House. Cardiff Road. Newport. Gwent NP9 1RH. Further copies may be obtained from Sales Branch. Unit 6. Nine Mile Point. Cwmfelinfach. Cross Keys. Newport. NP1 7HZ. Printed by Multiplex techniques ltd. St Mary Cray. Kent.